November 1991

LU TP 91-27
Harward Robotics Lab 91-20

Track Finding with Deformable Templates - The Elastic Arms Approach

Mattias Ohlsson[1] and Carsten Peterson[2]

Department of Theoretical Physics, University of Lund
Sölvegatan 14A, S-22362 Lund, Sweden

Alan L. Yuille[3]
Division of Applied Science, Harvard University
Cambridge, MA 02138 USA

Published in **Computer Physics Communications** 71 (1992) 77-98

Abstract:

A novel algorithm for particle tracking is presented and evaluated. It is based on deformable templates that converge using a deterministic annealing algorithm. These deformable templates are initialized by Hough transforms. The algorithm, which effectively represents a merger between neuronic decision making and parameter fitting, naturally lends itself to parallel execution. Very good performance is obtained for both non-magnetic and magnetic tracks. For the latter simulated TPC tracks from the CERN DELPHI detector are used.

[1] mattias@thep.lu.se
[2] carsten@thep.lu.se
[3] yuille%gramian@das.harvard.edu

1 Motivation and Results

Particle physics contains many challenging feature recognition problems ranging from off-line data analysis to low-level experimental triggers. In particular for the next generation of accelerators (LHC, SSC) the availability of efficient pattern recognition algorithms that can be executed in real-time will be crucial. The event rate at these machines is expected to be of the order of one event per 10-100 ns. One class of feature recognition problems is track finding. This is a combinatorial optimization problem; given a set of signals reconstruct particle trajectories subject to smoothness constraints.

Artificial neural network (ANN) techniques, or variations thereof, have shown great power in finding good approximate solutions to difficult optimization problems [1, 2, 3]. In refs. [4, 5] a neural approach was pursued for the track finding problem with encouraging results with respect to solution quality for toy-sized problems. The basic idea is to assign a decision element (neuron) S_{ij} between two signals i and j which is equal to 1 if i and j are connected and 0 otherwise. An energy function is constructed in terms of S_{ij} such that smooth tracks with no bifurcations correspond to minima. In its "raw" form this approach requires N^2 degrees of freedom for N signals. This assumes full potential connectivity. In reality this is never the case due to the locality of the problem; a track cannot pass too many pad-rows without giving rise to a signal. Also at high energies the curvature is limited, which constrains the connectivity in $(\Delta\theta, \Delta\phi)$. In ref. [6] such realistic cuts on the degrees of freedom were made on real TPC data from the CERN ALEPH detector. The performance in terms of quality of this ANN algorithm turns out to be compatible with the conventional one used in the ALEPH detector. With regard to execution speed the ANN approach is a winner, in particular for high multiplicity events [6]. Another local neural network inspired approach is to have a rotor [7] associated with each signal interacting in such a way that smooth tracks [8] are promoted. This method, which still needs to be worked out in more detail, would only require N degrees of freedom.

Even though the neural approach [4, 5, 6] seems to work very well it may not be the optimal way to proceed for the particle physics track finding problem for the following reasons:

1. It only solves the combinatorial optimization part of the problem; assigns signals to tracks. In reality one also needs to know the momenta corresponding the tracks. In the neural approach one then has to augment the algorithm with some fitting procedure. It would be nice to have a algorithm that does both things simultaneously.

2. The neural approach is presumably more general than what is needed for this problem. The parametric form of the tracks are known in advance - straight lines or helices. The network has no prior information about this. However, in other applications with no prior knowledge of the parametric form of the tracks, the very versatile ANN approach is the way to go[4].

3. The number of degrees of freedom needed to solve the a N signal problem is large even with the connectivity restrictions imposed in refs. [5, 6]. For a problem with N signals and M tracks one should only need 0(M) degrees of freedom.

4. As demonstrated in ref. [9] the neural approach is somewhat sensitive to noise. Again with prior knowledge of the parametric form one should be more robust with respect to noise.

All these issues can be accounted for in a novel approach [10] based on so-called deformable templates [11]

[4] In neural approaches there is of course always some prior knowledge or bias introduced in terms of penalty terms in energy functions.

or elastic nets [3]. A very similar approach was independently pursued in ref. [9]. The strategy is to try to match the observed events to simple parameterized models where the form of the models contains the *a priori* knowledge about the possible tracks - circles passing through the origin (the collision point).

Tracking elementary features and grouping them coherently is not a problem unique to the particle physics. It is an important problem in computer vision in general. There are many additional applications, such as the detection of incoming aircraft, but also interesting perceptual phenomena, such as those explored by Gestalt psychologists [12, 13]. In some of these applications the parametric form is not known in advance. In those instances one is of course better off with the non-prejudiced pure neuronic formulation [4, 5]. The template approach is based on a global view on the track finding problem, which is in line with how humans solve the problem. On the other hand, conventional track finding algorithms in particle physics like the *roadfinder* [14] are based on local optimization.

The particle tracking problem consists of fitting smooth curves through a set of data points, corresponding to the location of sensor responses. An unknown subset of these data points correspond to sensor "noise" and should be unmatched. Our strategy is to match the observed events to simple parameterized models. Currently we assume that the possible tracks are circles passing through the origin (the collision point). This assumption is correct if the particles move in a constant magnetic field and with negligible ionization losses (such losses would lead to changes in the curvature). The approach can be modified if necessary to allow for more complex paths. The formulation allows for some data points, hopefully those corresponding to sensor noise, to be unmatched. The mechanism involved is closely related to redescending M-estimators used in Robust Statistics [15]. For a discussion on this connection see ref. [16].

Hough transforms [17] are used to provide initial conditions for the templates and to specify the number of templates required. Hough transforms are essentially variants of "histogramming" or "binning" techniques which have previously been applied to particle tracking [18].

It turns out that the deformable templates and the Hough transforms are not unrelated. In the low temperature limit one can show the Hough transform is a special case of the deformable template algorithm [10]. A somewhat opposite approach was taken by Gyulassy and Harlander [9] who started out from the Hough transform and then generalized it to elastic nets or deformable templates.

Deformable templates and Hough transforms complement each other nicely for this problem. The deformable templates, with a deterministic annealing algorithm, give accurate fine-scale matching and can decline to match certain points - but need a rough estimate for the number of tracks. Hough transforms would guarantee to get the correct answer for noiseless data and for an infinitely small boxes in parameter space - but will make errors otherwise. Hough transforms, however, can be used to hypothesize a number of possible tracks which can be verified, or rejected, by deformable templates.

Some analytical support for this strategy comes from the work of De Veaux [20] on fitting data to a mixture of two Gaussian distributions. His fitting criterion can be reformulated in terms of deformable templates. De Veaux proves mathematically that, for a sufficient amount of data, a binning technique will provide a rough estimate for the parameters of the Gaussians which is sufficiently accurate to guarantee convergence to the globally optimum solution for the deformable templates.

The basic underlying idea of the deformable templates (or *elastic arms*) approach was briefly presented in ref. [10]. In this paper we develop this approach further with respect to theoretical understanding, extensions and implementation issues along the following lines:

- A local variant of the Hough transform is introduced, which is less computational expensive than the standard one.
- Both the Hough transform and the elastic arms approaches are extended to three dimensions. It turns out that this gives better results with respect to disentangling tracks which are close in the projected $\hat{x}\hat{y}$-plane.
- Non-magnetic (straight) tracks with different origins are dealt with in a successful manner.
- We demonstrate that the elastic arms algorithm easily deals with situations when it is initialized with too many tracks from the Hough transform - extra tracks are either attracted to noise sparks or to other arms, which have settled. Both cases can be dealt with by a removal procedure after the algorithm has settled.
- An alternative derivation of the elastic arms approach based on performing maximum likelihood estimation is given.
- The role of the effective repulsion term (winner-takes-all) in the algorithm is well understood.
- Phase transition properties of the algorithm are discussed.
- When implementing the algorithm the domains of attraction for each arm can be limited to one hemisphere or less, which speeds up the computations.
- Generic prescriptions for setting the parameters regulating convergence of the algorithm are given.

This paper is organized as follows: Geometrical conventions and the metric used is found in Sect. 2. In Sect. 3 we describe the classic Hough transform and the cost-effective local variant used in our approach. Sect. 4 contains the derivation of the deformable templates (elastic arms) algorithm. We also demonstrate how it gives rise to the Hough transform as a special case. Implementation issues and practical hints are discussed in Sect. 5. together with numerical explorations using simulated DELPHI TPC data. Finally in Sect. 6 the reader finds a brief summary and outlook.

2 Geometry and Metric

2.1 Curved Tracks

Throughout this paper we work with a constant magnet field in the $\hat{z}$-direction, $\vec{B} = B\hat{z}$. Furthermore, we neglect energy losses - all tracks are helices in the $\hat{x}\hat{y}$-plane. In three dimensions a track a is thus a spiral emerging from the origin with emission angle θ_a in the $\hat{x}\hat{y}$-plane, curvature κ_a and a parameter γ_a governing the longitudinal (non-curved) direction. In terms of these parameters a spiral is given by

$$\begin{aligned} x &= \frac{1}{\kappa_a}\{\sin(\theta_a + \mathrm{sgn}(\kappa_a)t) - \sin\theta_a\} \\ y &= \frac{1}{\kappa_a}\{-\cos(\theta_a + \mathrm{sgn}(\kappa_a)t) + \cos\theta_a\} \\ z &= \gamma_a t \end{aligned} \tag{1}$$

where $t \in [0, \pi]$ corresponding to a half-spiral (this limitation, which corresponds to real life, is necessary in order to avoid closed circles in the projected plane). We use κ rather than $r = 1/\kappa$ to describe the curvature since the former is bounded in track-finding situations where the detector defines a lowest possible helix radius.

The minimal *squared* Euclidean distance M_{ia} between a spiral a and a signal i with position (x_i, y_i, z_i) is given by

$$\begin{aligned} M_{ia} &= M_{ia}^{(xy)} + M_{ia}^{(z)} \\ &= \frac{1}{\kappa_a^2}\left\{1 - \sqrt{(\kappa_a x_i + \sin\theta_a)^2 + (\kappa_a y_i - \cos\theta_a)^2}\right\}^2 + (z_i - \gamma_a t)^2 \end{aligned} \tag{2}$$

where

$$t = -\mathrm{sgn}(\kappa_a)\left\{\mathrm{atan}\left(\frac{\kappa_a x_i + \sin\theta_a}{\kappa_a y_i - \cos\theta_a}\right) + \theta_a\right\} \tag{3}$$

We will later also apply the method to situations, where the events have been projected on to the $\hat{x}\hat{y}$-plane (cf ref. [10]). In those cases, of course, only the first term in eq. (2) is kept ($M_{ia}^{(xy)}$).

2.2 Straight Tracks

One might also want to consider situations with no magnetic field - straight tracks. In our case these applications, which are limited to two dimensions, are of two kinds. One is as above with a given vertex, in which case one has one parameter only, the emission angle θ_a in the equations describing the track. The other application is when the vertex (x_0, y_0) is also an unknown to be determined. The general equations for this read

$$\begin{aligned} x &= x_0 + t\cos\theta_a \\ y &= y_0 + t\sin\theta_a \end{aligned} \tag{4}$$

where $t \geq 0$.

3 The Hough Transform

As a first step we need a method that provides us with the approximate number of tracks in an event and the approximate parameter values for these arms. The Hough transform [17, 18] is appropriate for this. In its original form one determines a curve in parameter space for a given signal (x_i, y_i, z_i) corresponding to all possible tracks with a given parametric form it could possibly correspond to. All such curves belonging to the different signals are "drawn" in parameter space. This space is then discretized and entries are histogrammed - one divides parameter space up into boxes and counts the number of curves in each box. If the towers in the histogram exceeds thresholds in the parameter values $(\tilde{\theta}, \tilde{\kappa}, \tilde{\gamma})$ then the corresponding parameter values define a potential track. Needless to say, if the granularity of this procedure is chosen to be very small the computational requirements grows very rapidly. As will be shown in the next section this Hough transform can be derived as a limit of the elastic arms algorithm.

In ref. [10] a variant of the Hough transform was developed where all possible pairs of signals are picked and the intersection of their curves in parameter space is found. One of the parameters is then eliminated and the remainder(s) form the basis for a new histogram, which is has a cleaner signal-to-background separation than the original Hough transform. This version was found to be very successful with respect to solution quality but at the expense of heavy CPU demand.

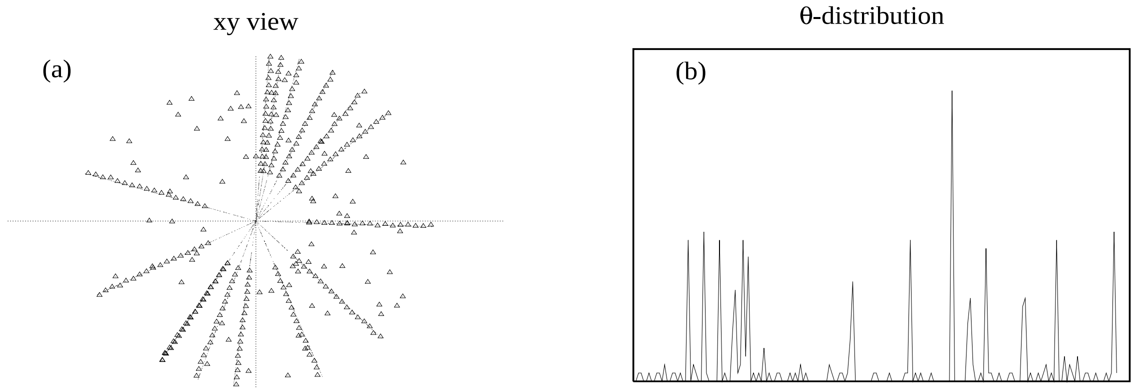

Figure 1: **(a)**. Generated signals corresponding to non-perfect straight tracks with noise together with the solution obtained from the local Hough transform. **(b)**. Resulting θ-distribution.

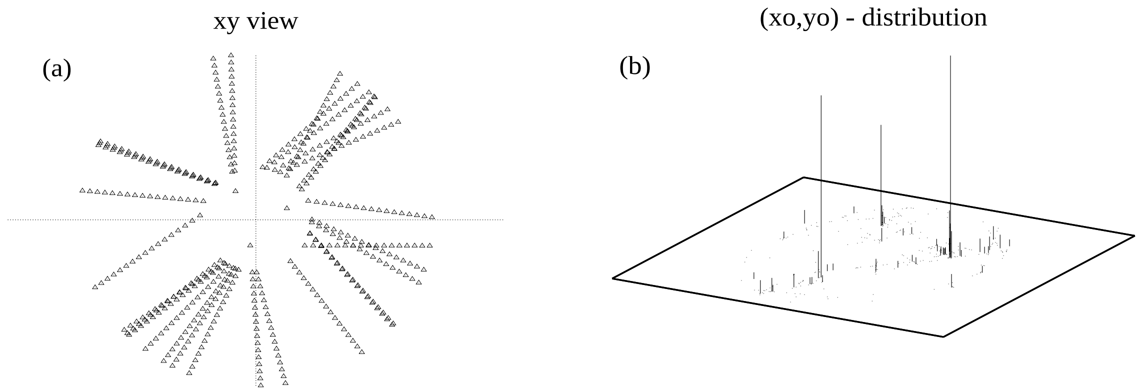

Figure 2: **(a)**. Generated signals corresponding to straight tracks emitted from three different vertices. **(b)**. Resulting (x_0, y_0)-distribution.

In this paper we present another alternative, which is local and hence very fast to execute. One defines a small neighbourhood (circle) around each signal and calculates the parameter values (κ_a, θ_a and γ_a) for each other signal within this neighbourhood. In this process one ignores signals sitting on the same

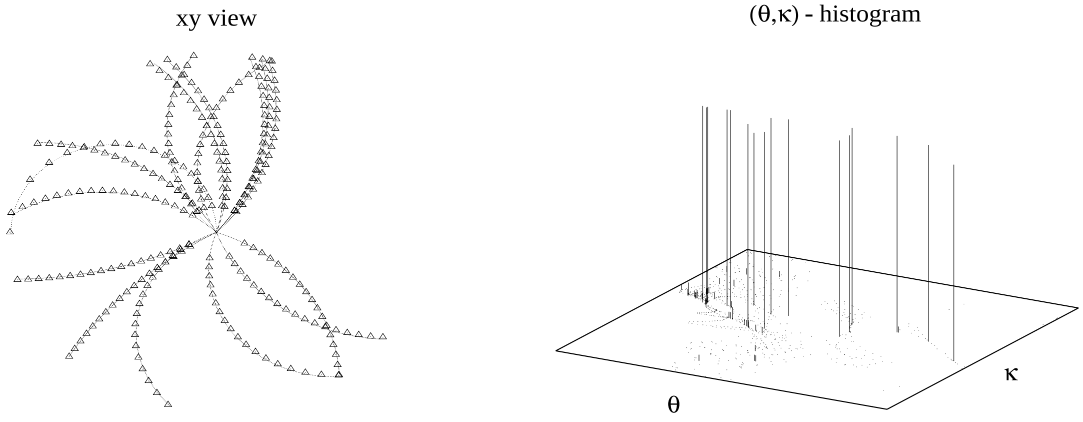

Figure 3: **(a)**. Generated signals corresponding to perfect helices with no noise together with the solution obtained from the local Hough transform. **(b)**. Resulting (θ, κ)-distribution.

pad-layer, since it is most unlikely that a track follows a pad-layer[5]. This procedure makes the number of calculations go like $O(N)$ and not like $O(N^2)$ when one takes all possible pairs of tracks. And, maybe more importantly, it reduces the noise. In what follows we use this *local Hough transform*. We have explored this technique in a number of different situations ranging from "academic" tracks to more realistic data including energy losses (non-perfect spirals) and noise signals.

Straight tracks. For perfect lines with no noise signals the Hough transform is of course trivial - it is an analytical transformation. In order to challenge the algorithm we have generated tracks with noise and small deviations from pure lines originating from the same vertex. In fig. 1a we show such an event. The resulting θ-distribution is shown in fig. 1b. As can be seen from fig. 1 the transform gives good solutions to the problem. No additional ambiguities need to get resolved by further processing. We have also generated signals corresponding to straight tracks (perfect lines with no noise) originating from three different vertices. Such an event is depicted in fig. 2 together with the parameter distributions underlying the solution. Again the algorithm seems to work very well. This example is particularly interesting since it gives a hint on how to handle situations involving vertex reconstruction and situations with multiple events as expected at LHC and SSC (non-perfect helices) and noise. Such an event projected onto the $\hat{x}\hat{y}$-plane is shown in fig 4 together with the (θ, κ)-distribution obtained with the local Hough transform.

To improve the solutions for straight tracks with the elastic arms method described in the next section is straightforward. We therefore leave straight tracks and move to curved ones.

Curved Tracks. Academic curved tracks with perfect helices and no noise signals are easily handled by the local Hough transform as is shown in fig. 3. This is contrast to "real" curved tracks generated by CERN DELPHI TPC event generator [19], which contain energy losses (the physical dimensions of the TPC detector can be found in section 5.2). As can be seen from fig. 4 the local Hough transform cannot solve the problem. Furthermore as can be seen from the (θ, κ)-distribution it is very difficult to have a unique procedure for defining a cut that yields the number of tracks. The origin of these problems are

[5] Pad-layers are concentrical rings of detector elements.

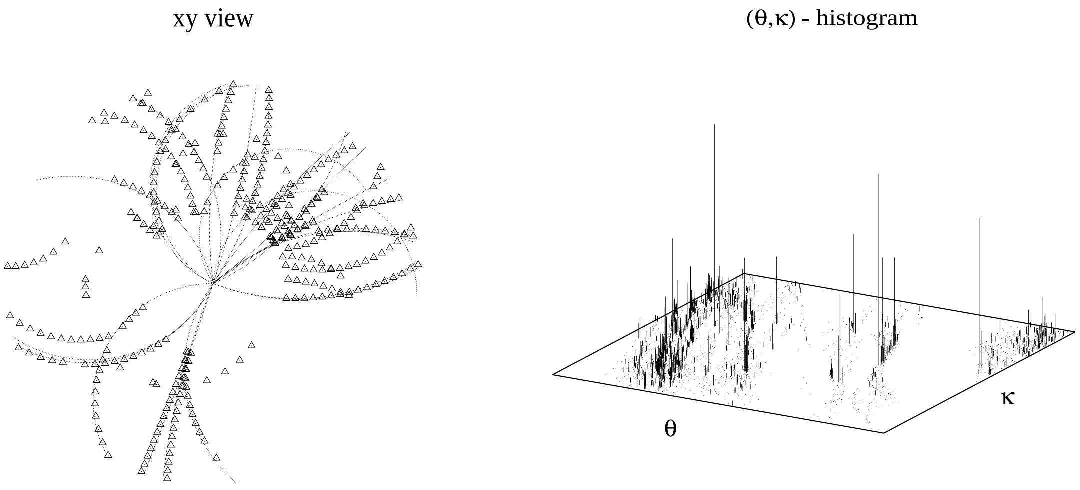

Figure 4: **(a)**. Signals generated by the CERN DELPHI TPC event generator together with the solutions generated by the local Hough transform. **(b)**. Resulting (θ, κ)-distribution.

threefold:

- Energy losses (non-perfect helices).
- Presence of noise signals.
- Decaying particles.

In a certain sense decaying particles (secondary vertices) is also a form of noise within our approach, since all trajectories are in general assumed to originate from a common origin. The first two items will be efficiently dealt with by the elastic arms approach, which is presented in the next section. Practical problem with extra vertices originating from decaying particles is not the focus of this paper - possible solutions are discussed in the end.

4 The Elastic Arms Approach

4.1 Derivation of the Algorithm

Armed with a set of M deformable templates $(\theta_a, \kappa_a, \gamma_a)$, $a = 1, ...M$, we fit them to the measured points (x_i, y_i, z_i) with a fitness measure defined as

$$E[V_{ia}; \theta_a, \kappa_a, \gamma_a] = \sum_{i,a} V_{ia} M_{ia} + \lambda \sum_i \{\sum_a V_{ia} - 1\}^2 \tag{5}$$

where V_{ia} is a binary decision unit such that

$$V_{ia} = 1 \tag{6}$$

if the a^{th} arm goes through the i^{th} point and is zero otherwise.

We want to minimize $E[V_{ia};\theta_a,\kappa_a,\gamma_a]$ with respect to V_{ia},θ_a,κ_a and γ_a subject to the global constraint that each point is either matched to a unique circle or not matched. More precisely, given i there exist a unique a such that $V_{ia}=1$ - a measured point should only be assigned to one arm. The second term in eq. (5) imposes a penalty λ if a specific point is unmatched to any circle. This can be related to Robust Statistics [15]. With this parameter λ one can allow for noise points not to be associated with any arm (e.g. cosmic ray events). Also it governs the finite width of a track in terms of signals being slightly displaced from an ideal arm. When finding global minima of an energy function one often needs to introduce noise in order to avoid getting stuck in local minima. A common procedure for this is *simulated annealing* [21] where the system is allowed to thermalize for a sequence of temperatures (noise) $T_n > T_{n-1} > ... > T_0$ according to the Boltzmann distribution

$$P[V_{ia};\theta_a,\kappa_a,\gamma_a] = \frac{1}{Z}e^{-\beta E[V_{ia};\theta_a,\kappa_a,\gamma_a]} \tag{7}$$

where $\beta = 1/T$ is the inverse temperature and Z is a normalization constant, the so-called partition function

$$Z = \sum_{\{V_{ia}\}}\sum_{\theta_a\kappa_a\gamma_a} e^{-\beta E[V_{ia};\theta_a,\kappa_a,\gamma_a]} \tag{8}$$

We now compute the so-called marginal probability distribution

$$P_M[\theta_a,\kappa_a,\gamma_a] = \sum_{\{V_{ia}\}} P[V_{ia};\theta_a,\kappa_a,\gamma_a] \tag{9}$$

by integrating (summing) out the neuronic degrees of freedom, V_{ia}. In doing this we must ensure that we sum only over configurations of the V's, which satisfy the global constraints defined above. As a first step we rewrite eqs. (7,9) as

$$P_M[\theta_a,\kappa_a,\gamma_a] = \frac{1}{Z}\sum_{\{V_{ia}\}}\prod_i e^{-\beta\sum_a V_{ia}M_{ia}-\beta\lambda\{\sum_a V_{ia}-1\}^2} \tag{10}$$

The different possible V_{ia}'s correspond to functions $a'(i)$ which are either 1 or 0. This means that the first term in the exponent of eq. (10) only gets contribution for $a'(i)$'s, which are 1. Correspondingly the second term is only non-zero when $a'(i)=0$. This gives

$$\begin{aligned} P_M[\theta_a,\kappa_a,\gamma_a] &= \frac{1}{Z}\sum_{a'(i)}\prod_i e^{-\beta M_{ia'(i)}-\beta\lambda} \\ &= \frac{1}{Z}\prod_i\sum_a e^{-\beta M_{ia}-\beta\lambda} \end{aligned} \tag{11}$$

where in the last step we have interchanged the order of the summation and product and made the notational replacement $a'(i)\rightarrow a$. We now rewrite eq. (11) as

$$P_M[\theta_a,\kappa_a,\gamma_a] = \frac{1}{Z}e^{-\beta E_{eff}[\theta_a,\kappa_a,\gamma_a]} \tag{12}$$

where we have introduced the *effective* energy E_{eff} as

$$E_{eff}[\theta_a,\kappa_a,\gamma_a] = -\frac{1}{\beta}\sum_i \log\{e^{-\beta\lambda}+\sum_a e^{-\beta M_{ia}}\} \tag{13}$$

We are looking for the most probable configurations to eq. (12). These should be given by the minima of E_{eff} with respect to θ_a, κ_a and γ_a. This is done by a gradient descent method for a sequence of decreasing temperatures (annealing). One gets

$$\Delta\theta_a = -\eta \frac{\partial E_{eff}}{\partial \theta_a} = -\eta \sum_i \hat{V}_{ia} \frac{\partial M_{ia}}{\partial \theta_a} \tag{14}$$

$$\Delta\kappa_a = -\eta \frac{\partial E_{eff}}{\partial \kappa_a} = -\eta \sum_i \hat{V}_{ia} \frac{\partial M_{ia}}{\partial \kappa_a} \tag{15}$$

$$\Delta\gamma_a = -\eta \frac{\partial E_{eff}}{\partial \gamma_a} = -\eta \sum_i \hat{V}_{ia} \frac{\partial M_{ia}}{\partial \gamma_a} \tag{16}$$

or in a more generic form with notation $\vec{\pi}_a = (\theta_a, \kappa_a, \gamma_a)$

$$\Delta\vec{\pi}_a = -\eta \nabla E_{eff} = -\eta \sum_i \hat{V}_{ia} \nabla M_{ia} \tag{17}$$

where ∇ operates in $\vec{\pi}_a$ space. In eqs. (14-17) the *Potts factor* $\hat{V}_{ia}$ is given by

$$\hat{V}_{ia} = \frac{e^{-\beta M_{ia}}}{e^{-\beta\lambda} + \sum_b e^{-\beta M_{ib}}} \tag{18}$$

From eq. (2) one easily computes the partial derivatives $\partial M_{ia}/\partial\theta_a$, $\partial M_{ia}/\partial\kappa_a$ and $\partial M_{ia}/\partial\gamma_a$, which can be found in appendix A.

4.2 A Bayesian View

Maximizing $P_M[\theta_a, \kappa_a, \gamma_a]$ can be thought of as performing a maximum likelihood estimation of the parameters $\theta_a, \kappa_a, \gamma_a$ of a model that generates data $\vec{r}_i$ with probability

$$P[\vec{r}_i : \theta_a, \kappa_a, \gamma_a] = \frac{1}{Z^*}\{e^{-\beta\lambda} + \sum_a e^{-\beta M_{ia}}\} \tag{19}$$

We then apply Bayes' theorem to obtain

$$P[\theta_a, \kappa_a, \gamma_a : \vec{r}_i] = \frac{P[\vec{r}_i : \theta_a, \kappa_a, \gamma_a] P_M[\theta_a, \kappa_a, \gamma_a]}{P[\vec{r}_i]} \tag{20}$$

If we make the reasonable assumption that the prior probability of κ_a, θ_a and γ_a, $P[\theta_a, \kappa_a, \gamma_a]$, is uniform then finding the best *a posteriori* estimate, maximizing $P[\theta_a, \kappa_a, \gamma_a : \vec{r}_i]$ with respect to $\theta_a, \kappa_a, \gamma_a$, is equivalent to maximizing $P[\vec{r}_i : \theta_a, \kappa_a, \gamma_a]$.

This corresponds to a mixture of distributions and can be thought of as multiple regression (see ref. [20]). The β-parameter must now be interpreted as a measure of the spread of the distribution and must be estimated from our sensor model. Our probability distribution therefore assumes that the data comes either from one of the templates $\theta_a, \kappa_a, \gamma_a$ or from a uniform distribution parameterized by λ. Note that if we allow the $\vec{r}_i$ to occur in an infinite range then, because of the uniform distribution, the normalization factor Z^* will be undefined. This need not concern us, however, since we are only interested in the most probable state.

4.3 How Does the Algorithm Work?

How does this algorithm work? At the starting temperature T_H a set of template arms are placed according to the Hough transform values for the parameters θ_a, κ_a and γ_a. The templates are Gaussian distributions centered around the arm values where the width is given by the temperature (see fig. 5). Initially each arm can attract many signals. The relative importance of the different signals is measured by the Potts factor (eq. (18)). As the temperature is lowered the different arms are attracted to nearby signals more intensely. As discussed in connection with eq. (5) λ governs the amount of noise points or *outliers* the algorithm

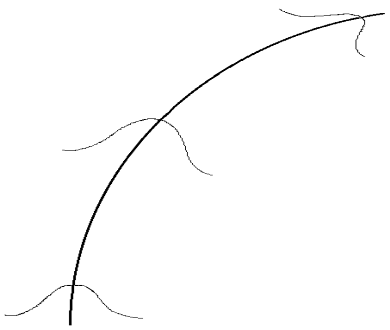

Figure 5: An elastic arm at temperature T.

allows for. It enters the Potts factor (eq. (18)) as a "zero" neuron in the denominator which is never updated - it does not depend upon M_{ia}. For $\lambda \to \infty$ no noise is ignored and the "zero" neuron vanishes. For finite values of λ the "zero" neuron absorbs all data outside $1/\lambda$ from the domain of attraction of the arms.

4.4 The Effective Repulsive Force

The Potts factor *implicitly* contains a repulsive force with its winner-takes-all structure. Consider a situation where an arm is located approximately in between two tracks of signals. The track which has smallest M_{ia}-values with respect to the arm (no matter how small the possible difference is) will take over and move the arm to the correct position. Any other arm in the "area" will be attracted to other tracks. In this sense there is a repulsive force between the arms. This repulsion force is transparent already at the effective energy level. Omitting an unimportant $\exp(\lambda\beta)$ factor eq. (13) reads

$$E_{eff}[\theta_a, \kappa_a, \gamma_a] = -\frac{1}{\beta}\sum_i \log\{1 + \sum_a e^{-\beta(M_{ia}-\lambda)}\} \tag{21}$$

Taking the limit as $\beta \mapsto \infty$ gives

$$\begin{aligned} \lim_{\beta\mapsto\infty} \frac{1}{\beta}\log\{1 + \sum_a e^{-\beta(M_{ia}-\lambda)}\} \quad &\mapsto \quad 0, \ \text{ if } M_{ia} > \lambda \text{ for all a} \\ &\mapsto \quad \lambda - \min_a M_{ia} \ \text{ otherwise} \end{aligned} \tag{22}$$

If the algorithm is initiated by too many arms from the Hough transform the extra tracks will behave according to the following 3 possibilities:

1. Attracted to noise.

2. Attracted to points belonging from tracks originating from decay vertices.

3. Attracted to a track upon which another arm has already settled in .

In the third alternative the effective repulsive force dies away when one of the arms is already settled. One gets an extra arm on top of another. This is in contrast to the approach of ref. [9], where one-arm templates are fitted to the data in a serial way. Incrementally fitted arms avoid already fitted signals by an *explicitly* introduced repulsive interaction term $w^2/(w^2+x^2)$, which is quite similar to the one in eq. (21) - both are unimodal and peaked at the origin, and fall to zero at infinity. For large values of β varying w is essentially the same as varying λ. In our terminology, therefore, Gyulassy and Harlander's strategy [9] of decreasing w during the computation corresponds to reducing λ and is an alternative to annealing on β.

4.5 Phase Transition Properties

Decision problems described in thermodynamical language typically contain phase transitions when going from high to low temperatures. This is the case for problems that can be mapped onto pure neuronic magnetic-like systems [2], but also for template systems like the elastic net for the traveling salesman problem [3, 23] and in principle also for this one. In the traveling salesman case a set of template cities with spring-force interactions are placed on circle with an origin close to the the center of gravity of the signals (in this case the cities). The equation have a similar form to those of eqs. (14,15,14), the main difference being the absence of the "zero" neuron. As the temperature is lowered the circle containing the template cities is deformed and expanded to match the cities. During this annealing the corresponding Potts factors sharply go from all components being equal to a situation where one component is 1 and the remainders 0. Such a phase transition behaviour is often conveniently monitored by the *saturation* $\Sigma = 1/N \sum_{a,i} \hat{V}_{ia}^2$ [2], which rises sharply from $1/M$ to 1, where M is the number of template cities, when the temperature goes from infinity down to zero. In the traveling salesman the $T \to \infty$ corresponds to the circle being contracted to a point where the distances to all cities are the same (center of gravity).

What is the corresponding $T \to \infty$ limit in our case of track finding? It corresponds to a situation where all the arms have identical parameters given by minimal distance to all signals. In principal one could initialize the algorithm in the vicinity of this trivial fixed-point. However, from the point of view of computational speed it is more advantageous to use the Hough transform parameters values as a starting point - this means that the algorithm is initialized below the phase transition point T_c. Being initialized below T_c does not mean that all matching decisions are taken by the Hough transform. It only has impact on the global distribution of arms - many matching decisions between nearby and crossing tracks needs to be taken. This is efficiently done by the elastic arms algorithm.

4.6 Relation to EM Algorithms

An alternative algorithm for minimizing eq. (5) is the EM algorithm [24] which proceeds iteratively by alternating two operations. The first minimizes $E[V_{ia};\theta_a,\kappa_a,\gamma_a]$ with respect to the $\theta_a,\kappa_a,\gamma_a$ variables with the V_{ia} being fixed. The second then calculates the best estimate of the V_{ia} analytically using eq. (18). EM algorithms[6] have empirically shown themselves to be very quick to converge though there is currently no theoretical results which explain this.

4.7 The Hough Transform Limit

There is an interesting relation between the effective energy for our deformable templates and the Hough transform. Whereas the authors of ref. [9] start with the Hough/Radon transform [7] and generalize it to elastic tracking our approach implies the opposite. We start with deformable templates and show that the Hough transform comes out in the large β small λ limit.

The effective energy for *one* deformable template is given by ($a = 1$)

$$E_{eff}[\alpha] = -\frac{1}{\beta}\sum_i \log\{e^{-\beta\lambda} + e^{-\beta M_{i1}(\alpha)}\} \tag{23}$$

where $\alpha = (\theta_1,\kappa_1,\gamma_1)$ are the parameters of the single template and $M_{i1}(\alpha)$ is the closest distance between the data point $\vec{r}_i$ and the template parameterized by α.

Minimizing $E_{eff}[\alpha]$ with respect to α is equivalent to maximizing

$$H[\alpha:\beta,\lambda] = \frac{1}{\beta}\sum_i \log\{1 + e^{-\beta(M_{i1}(\alpha)-\lambda)}\} \tag{24}$$

where we have dropped a constant factor $N\lambda$ from the cost function.

In the limit as $\beta \mapsto \infty$ we find

$$\begin{aligned} \frac{1}{\beta}\log\{1 + e^{-\beta(M_{ia}-\lambda)}\} &\mapsto 0, \quad \text{if } M_{i1} > \lambda, \\ &\mapsto \lambda - M_{ia}, \quad \text{if } M_{ia} < \lambda \end{aligned} \tag{25}$$

This already has the flavour of a histogramming technique; if a point $\vec{r}$ is sufficiently close (closeness is measured by λ) to a curve with parameters α then it gives a contribution to $H[\alpha:\beta,\lambda]$. The discrete limit can be attained by dividing $H[\alpha:\beta,\lambda]$ by λ. Then, as $\lambda \mapsto 0$, we get a contribution $= 1$ if $M_{ia} = 0$ and zero otherwise - this is nothing but the Hough transform. The same argument is easily generalized to the continuous limit, the Radon transform [10].

Thus we obtain the Radon/Hough transform in the limit as $\beta \mapsto \infty$ and $\lambda \mapsto 0$. In this limit we can use standard Hough techniques to find the extrema of $H[\alpha:\beta,\lambda]$, alternatively we could use gradient descent algorithms with multiple starting points. This is not entirely surprising. The $\beta \to \infty$ limit corresponds to

[6] EM is short for having iterations where an *expectation* step is followed by a *maximization* step.

[7] The Radon transform is the continuous version of the Hough transform.

$T \to 0$, which is gradient descent in neural net language. Multiple gradient descents from different seeds is a form of exploratory search which the Hough transform does in the zero resolution limit. In ref. [22] it is demonstrated that for the related problem of image segmentation a number of existing algorithms can similarly be understood in terms of "annealing" along different parameters.

5 Simulations and Results

5.1 Implementation Issues

5.1.1 The Hough Transform

The Hough transform is used to initialize the elastic arms algorithm. In its standard form possible track parameters $\vec{\pi}_a$=$(\theta_a, \kappa_a, \gamma_a)$ with finite resolutions $\Delta\theta_a$, $\Delta\kappa_a$ and $\Delta\gamma_a$ are determined from the measured signals $\vec{r}_i$ by solving the equation

$$M_{ia}(\vec{r}_i, \pi_a) = 0 \tag{26}$$

One then makes a histogram in parameter space to find the most "popular" parameter values within the resolution. Eq. (26) has very many solutions[8] for every $\vec{r}_i$. This fact together with the existence of energy losses (non-perfect helices) and of noise makes the standard Hough transform not so effective. We therefore use a local Hough transform (see sect. 3) which consists of two steps; first θ_a and κ_a are determined using signals projected onto the $\hat{x}\hat{y}$-plane and then γ_a are determined using the found θ_a and κ_a.

1. **Projected tracks.** First we define a circle with radius ρ_{xy} around each signal i such that it covers at least two pad-layers in the detector. For all signals j within the ρ_{xy}-neighbourhood except those belonging to the same pad-layer we then simultaneously solve

$$\begin{aligned} M_{ia}^{(xy)} &= 0 \\ M_{ja}^{(xy)} &= 0 \end{aligned} \tag{27}$$

   for all pairs of projected signals $\vec{r}_i^{(p)}$ and $\vec{r}_j^{(p)}$. The solution to eq. (27), which can be found in appendix B, is a helix going through $\vec{r}_i^{(p)}$, $\vec{r}_j^{(p)}$ and (0,0). These solutions are then used to make a histogram in (θ, κ)-space. The size of the detector defines the upper limit of κ $(0 \leq |\kappa| \leq 1/r_{min})$, where r_{min} is the the distance from the collision vertex to the inner boundary of the cylindrical detector. In cases where one is only interested in particles with a certain minimal energy the upper limit of κ should be lowered accordingly. The emission angle limits $(0 \leq \theta_a \leq 2\pi)$ can of course also be changed if one is only interested in particles appearing in certain directions. The entries (θ_a, κ_a) that exceed certain thresholds are then kept as potential (projected) tracks for the next step.

2. **Correlation with the longitudinal dimension.** We now use the established set of $\{\theta_a, \kappa_a\}$ to see if there is a correlated set $\{\gamma_a\}$. Again this can be done locally by defining a small neighbourhood ρ_z around each found track in the $\hat{x}\hat{y}$-plane - ρ_z defines a thin "sausage" around the track (for the DELPHI TPC data we chose ρ_z=5). One track (θ_a, κ_a) is chosen and the equation

$$M_{ia}^{(z)} = 0 \tag{28}$$

[8] In the continuum limit a circular track corresponds to infinitely many solutions.

is solved for each signal $\vec{r}_i$ that has a projection $\vec{r}_i^{(p)}$ in this neighbourhood. One then makes a histogram in γ-space to find the most popular γ. If (θ_a, κ_a) is a valid track then there should only be one substantial peak in the histogram corresponding to its γ_a-value. If no such distinct peak exists, it is very likely that the track (θ_a, κ_a) is nonvalid one. These last two steps are repeated for every (θ_a, κ_a)-pair.

We are now armed with a set of spirals $\vec{\pi}_a = (\theta_a, \kappa_a, \gamma_a)$ to initiate the elastic arms algorithm with.

5.1.2 The Elastic Arms Algorithm

Given the approximate number of arms and the corresponding initial values of parameters from the Hough transform we next minimize the effective energy $E_{eff}[\theta_a, \kappa_a, \gamma_a]$ defined in eq. (13) using the gradient descent equations (eqs. (14-17)). In this subsection we give a set of prescriptions and hints of how to ensure good and rapid convergence in a way that is as problem independent as possible.

From eq. (17) we have $\Delta\vec{\pi}_a = -\eta\nabla E_{eff}$. However, the partial derivatives $E_{eff}/\partial\theta_a$, $\partial E_{eff}/\partial\kappa_a$ and $\partial E_{eff}/\partial\gamma_a$ all have different magnitudes. We therefore use different update rates for these different parameters. Also the nature of κ is different than the other two. For an almost straight track a minor change in κ has more impact than on a strongly curved track. For this reason we need learning rates for κ that depends strongly upon $|\kappa|$. This naturally implies individual κ update rates, $\eta_\kappa^{(1)}$,$\eta_\kappa^{(2)}$,...,$\eta_\kappa^{(M)}$. For θ_a and γ_a the situation is the opposite since they should be independent of $|\theta_a|$ and $|\gamma_a|$. We therefore have common update rates η_θ and η_γ for these parameters. In summary we have:

- Common update rates η_θ and η_γ for all M arms.
- Individual update rates $\eta_\kappa^{(a)}$ for each arm a.

Also, the magnitude of the partial derivatives depend strongly upon the magnitude of our signal coordinates $|\vec{r}_i|$. In order to make the update rates less dependent on different tracking-problems we rescale the signals $\vec{r}_i$ to some predefined dynamic range, in our case, $|\vec{r}_i| \leq 10$.

When initiating the η's we want a smooth transition from the the Hough parameter values. This can be accomplished by choosing η_θ, η_γ such that $\eta_\theta(1/M)\sum_a |\partial E_{eff}/\partial\theta_a|$ and $\eta_\gamma(1/M)\sum_a |\partial E_{eff}/\partial\gamma_a|$ are small numbers as compared to the range of θ and γ respectively. The η_κ's are chosen such that $\Delta\kappa_a$ is some fraction of κ_a, where we again as an estimate of $|\partial E_{eff}/\partial\kappa_a|$ use $(1/M)\sum_a |\partial E_{eff}/\partial\kappa_a|$.

As mentioned earlier, T is measure of the width of the Gaussian around each arm, therefore T_0 and T_{final} should be chosen with respect to the magnitude of the dynamical range.

The left hand sides of eqs. (14-16) in principle contain sums over all signals i. But the template arms are in reality half-spirals which means that the sums should be restricted to include only those signals lying in the same half-sphere as the arm does. If we are looking for very high energy particles with small curvatures this aperture can be limited even more leading to a significant speed-up of the algorithm.

As mentioned in section 4.4 it is possible for extra tracks to be attracted to noise-signals. These can be removed after the algorithm if we require valid tracks to pass through a minimal number of signal-points.

1. Obtain an initial set of arms from the local Hough-transform.

2. Rescale the signals $\vec{r}_i$ to the dynamical range.

3. Choose update rates η_θ, η_κ, η_γ and α according to the dynamic range. Do the same thing for λ, T_0 and T_{final}.

4. For a sequence of temperatures $T_n = kT_{n-1}$, $T_n \leq T_{final}$, with k=0.95, update according to eqs. (14-16,29).

5. Make it converge at $T = T_{final}$ by lowering the update rates until E_{eff} is not changing.
   $\eta_\theta = \epsilon \times \eta_\theta$
   $\eta_\kappa^a = \epsilon \times \eta_\kappa^a$
   $\eta_\gamma = \epsilon \times \eta_\gamma$
   where ϵ=0.9.

6. Delete extra tracks, which are attracted to noise or double ones (see text).

Figure 6: The elastic arms algorithm for particle-tracking. The algorithm is not very sensitive to choice of k and ϵ. Parameters specific for the DELPHI TPC experiment can be found in table 1.

Another possibility is that the extra track becomes identical with an already existing track. This check should be done after the algorithm has converged.

The parameter λ governs the relative importance of signals not associated with any track. Since there are noise-signals and signals originating from secondary vertices (also considered as noise in this context) λ should be kept small as compared to an average M_{ia}, allowing signals not to be matched to any track.

The gradient descent method is just one way of minimizing E_{eff}, which is very simple to implement. Other, more elaborate and powerful minimization procedures involves the second derivative (Hessian Matrix) of E_{eff}. Conjugate gradient descent is another possible minimization procedure where only the first derivative of E_{eff} is needed [25]. We did not use these methods in our application studies. A very simple way to improve the gradient descent method is to introduce a so-called momentum term. Each degree of freedom π_a is given some inertia or momentum. In other words $\Delta\pi_a(t)$ gets a contribution from $\Delta\pi_a(t-1)$ according to

$$\Delta\vec{\pi}_a(t) = -\vec{\eta}\nabla E_{eff} + \alpha\Delta\vec{\pi}_a(t-1) \tag{29}$$

where $0 < \alpha < 1$. This means that p_a feels an *average* downhill "force" when moving on the energy surface. The momentum term can prevent the energy from oscillating and hence make the minimizing more effective. We use this updating applications in our with $\alpha = 0.5$.

In fig. 6 we show a "black box" prescription of the elastic arms algorithm that we use in our application studies.

Local Hough transform	resolution $(\Delta\theta, \Delta\kappa)$	[180,100]
	threshold $(\tilde{\theta}, \tilde{\kappa})$	6
Elastic Arms	dynamic range	$\|\vec{r}_i\| \leq 10$
	λ	0.01
	T_H	0.2
	T_{final}	0.01
	ϵ	1.06
	α	0.5

Table 1: Detector specific parameter values for the local Hough and the elastic arms algorithm using data from the CERN DELPHI TPC detector.

.

5.2 Numerical Explorations

We have tested the performance of the algorithm with simulated data from the CERN DELPHI TPC detector [19]. This detector is a cylinder with 35 cm inner and 111 cm outer radius and a length of 2.7 m. This simulator is supposed to very realistic with respect to energy losses, noise etc.. Hence it should be sufficient as a challenging test bed for our algorithm. The elastic arms algorithm we used follows closely the one in figure 6. Detector specific parameters choices are found in table 1. The final results are shown

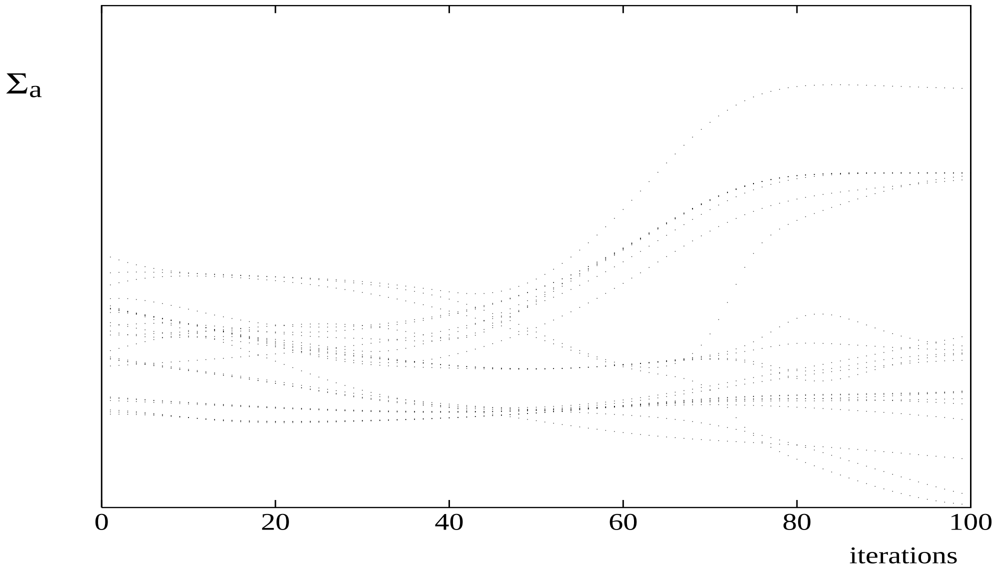

Figure 7: Development of the Σ_a's. These have been normalized to the average number of signals per track ($\approx$16).

in figures 8 and 9. It is encouraging to see how well the algorithm works. The arms do not confuse one another, even when passing close, or crossing each other. This is of course due to the Potts factor (eq. (18)), which in a sense "decides" which track each arm should be attached to and ignores the others. It is interesting to see how different $\Sigma_a = \sum_i \hat{V}_{ia}$ develop with decreasing temperature (or iteration step). This is shown in fig. 7. In this figure we have deliberately chosen to initiate the algorithm at a high starting value for the temperature. This implies that after a few iterations some arms will become identical (see section 4.5). During the annealing process different arms are then attracted to different tracks. This can be seen in fig. 7 as the development of Σ_a's. There are also "neurons" that die out, which means that they are redundant and do not correspond to tracks. In practice, however, we use a lower initial temperature since we do not want to "destroy" the Hough initialization. This decreases the number of iterations needed to be approximately 40-50 for DELPHI TPC problems.

When the algorithm is initiated at a temperature close to the one corresponding to the Hough transform values the problem is easier in the sense that many of the rough Hough estimates of the track parameters only need fine tuning. It is impressive that the algorithm is also able to solve problems when this "intelligent" initialization is smeared out at a higher initial temperature as is the case in fig. 7.

Errors that may occur comes from tracks that violate the assumptions for this algorithm, that is, (i) tracks that not originate from our *a priori* known vertex position and (ii) tracks that, due to energy losses, are not spirals. The problem with secondary vertices can, in principle, be dealt with if we introduce a new parameter $\vec{r}_a^{(o)}$, which is the vertex position for track a. The elastic arms algorithm, derived in section 4.1, does not change, but the distance measure M_{ia} is now also a function of $\vec{r}_a^{(o)}$. A generalization to include unknown vertex positions should be straightforward. Problem (ii) comes from particles with very low energy, and usually one wants to ignore these kind of particles. If, for some reason, one wants to to detect these low energy particles one would have to use a new parameterization for the arm to include energy losses.

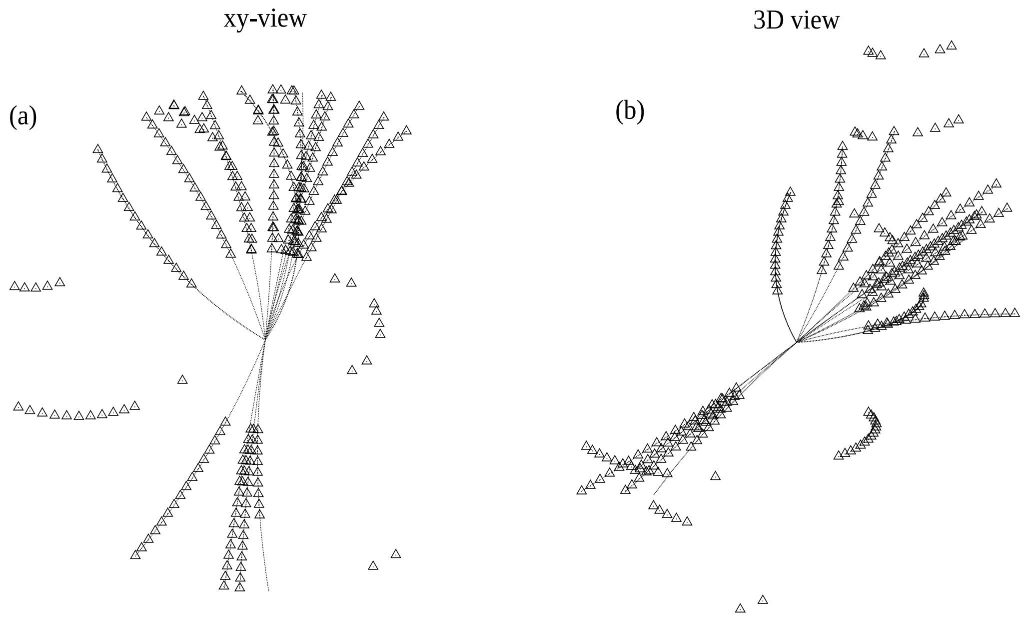

Figure 8: **(a)**. Result from Hough/Elastic arms algorithm with signals generated by CERN DELPHI TPC event generator (308 signal points). **(b)**. The same result projected onto the $\hat{x}\hat{y}$-plane.

We consistently find that treating the problem in three dimensions with our algorithm is crucial when resolving some assignment ambiguities appearing on the two-dimensional level.

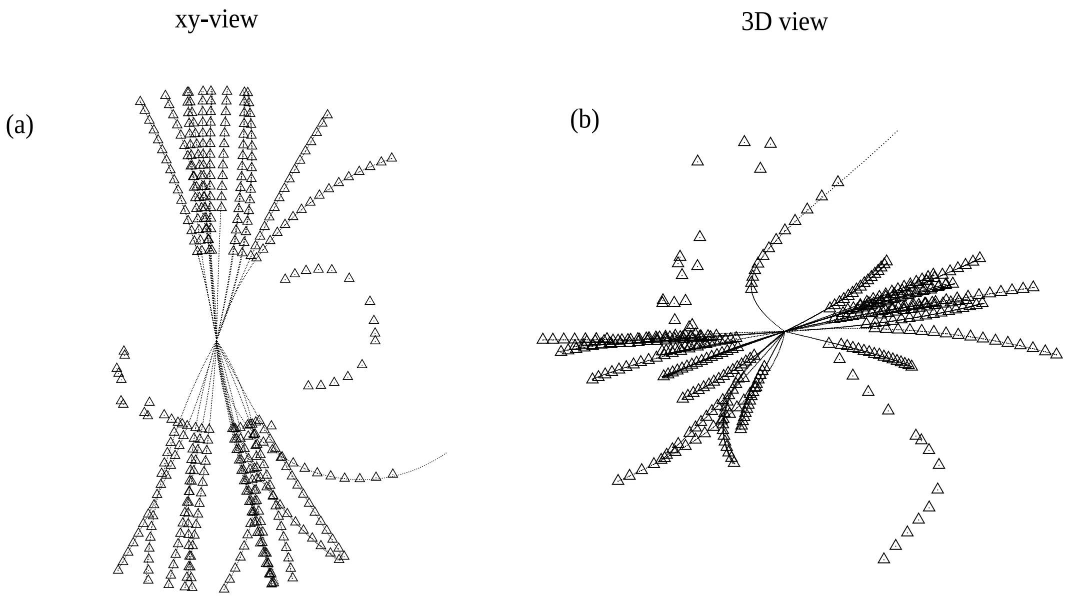

Figure 9: **(a)**. Result from Hough/Elastic arms algorithm with signals generated by CERN DELPHI TPC event generator (395 signal points). **(b)**. The same result projected onto the $\hat{x}\hat{y}$-plane.

Not only does the algorithm exhibit good performance, it is also very cost effective in every respect. In table 2 we show scaling properties and time consumption. The underlying F77 code contains only $O(300)$ lines.

	Scaling Properties	CPU consumption [DEC3100]
Hough transform	N -N^2	$O(1 \text{ sec})$
Elastic Arms	N$\times$ M	$O(1 \text{ min})$

Table 2: General scaling properties and time consumption needed to a process a typical DELPHI TPC event. N and M denote the number of signals and potential arms respectively.

6 Summary and Outlook

We have devised a track finding method that combines the matching and the fitting problem into a single algorithm. It goes from coarse to fine resolution by using a variant Hough transform to initialize a set of elastic arms. The latter settle in an annealing process in a deterministic way to deliver the final parameters - the momenta.

The approach gives rise to high quality solutions with very good scaling properties (approximately linear with the number of signals) and modest CPU consumption. It is straightforward to implement on a parallel processor. The algorithm is fairly insensitive to convergence parameters - we have applied it to DELPHI TPC data in a "black box" manner.

The elastic arms approach is very similar to human processing for this kind of recognition problem. A human looks for helices in a global way and then makes fine-tuning adjustments. This is in contrast to conventional roadfinder methods [14] and pure neuronic approaches [4, 5], which are based on more local considerations.

The algorithm is closely related to robust statistics - it ignores noise to a desired level.

The approach is easy to adapt to specific situations. For example, suppose measurement precisions vary for different pad-layers. Then the formalism can be generalized to allow for different i-dependent λ's for the different pad-layers.

In this paper we have focused on the basics of the algorithm and methology for getting fast convergence. Our main application study was simulated curved tracks (including noise) in the DELPHI TPC detector. At this point we ignored secondary vertices from decaying particles. These can be accommodated by allowing for more parameters describing the arms (vertex positions). Preliminary studies of this extension using straight tracks looks promising. Also with these extensions the algorithm could be used for vertex detection in general. At LHC/SC luminosities one expects multiple events per bunch crossing. Again, allowing for extra parameters, this algorithm has the potential of disentangling such events. Cerenkov images are also of parametric nature and should hence be tractable with this approach.

Acknowledgements

We would particularly like to thank O. Barring for providing us with the DELPHI TPC simulation data. We would also like to thank K. Honda for discussions and to acknowledge support from DARPA with contract AFOSR-89-0506.

Appendix A

In this appendix we give the partial derivatives computed from eq. (2) that are needed in the elastic arms equations (eqs. (14,15,16)).

$$\frac{\partial M_{ia}}{\partial \theta_a} = \frac{\partial M_{ia}^{(xy)}}{\partial \theta_a} - 2\gamma_a(z_i - \gamma_a t)\frac{\partial t}{\partial \theta_a} \tag{A1}$$

$$\frac{\partial M_{ia}}{\partial \kappa_a} = \frac{\partial M_{ia}^{(xy)}}{\partial \kappa_a} - 2\gamma_a(z_i - \gamma_a t)\frac{\partial t}{\partial \kappa_a} \tag{A2}$$

$$\frac{\partial M_{ia}}{\partial \gamma_a} = -2t(z_i - \gamma_a t) \tag{A3}$$

where t is found in eq. (3) and the $\hat{x}\hat{y}$-projected quantities are given by

$$\frac{\partial M_{ia}^{(xy)}}{\partial \theta_a} = -\frac{2}{\kappa_a}\left\{\frac{1}{\sqrt{(\kappa_a x_i + \sin\theta_a)^2 + (\kappa_a y_i - \cos\theta_a)^2}} - 1\right\}(x_i\cos\theta_a + y_i\sin\theta_a) \tag{A4}$$

$$\begin{aligned}\frac{\partial M_{ia}^{(xy)}}{\partial \kappa_a} &= -\frac{2}{\kappa_a^3}\left\{1 - \sqrt{(\kappa_a x_i + \sin\theta_a)^2 + (\kappa_a y_i - \cos\theta_a)^2}\right\}\times \\ &\times \left\{1 + \frac{\kappa_a y_i\cos\theta_a - \kappa_a x_i\sin\theta_a - 1}{\sqrt{(\kappa_a x_i + \sin\theta_a)^2 + (\kappa_a y_i - \cos\theta_a)^2}}\right\}\end{aligned} \tag{A5}$$

and

$$\frac{\partial t}{\partial \theta_a} = -\mathrm{sgn}(\kappa_a)\left\{\frac{\kappa_a y_i\cos\theta_a - \kappa_a x_i\sin\theta_a - 1}{(\kappa_a x_i + \sin\theta_a)^2 + (\kappa_a y_i - \cos\theta_a)^2} + 1\right\} \tag{A6}$$

$$\frac{\partial t}{\partial \kappa_a} = \mathrm{sgn}(\kappa_a)\left\{\frac{x_i\cos\theta_a + y_i\sin\theta_a}{(\kappa_a x_i + \sin\theta_a)^2 + (\kappa_a y_i - \cos\theta_a)^2}\right\} \tag{A7}$$

Appendix B

In order to calculate (θ_a, κ_a) for a helix going through $\vec{r}_i^{\,p}$, $\vec{r}_j^{\,p}$ and (0,0), we need to solve eq. (27) with M_{ia}^{xy} given by eq. (2). With (x_i, y_i)=$\vec{r}_i^{\,p}$, we get

$$\begin{aligned}\frac{1}{\kappa_a^2}\left\{1 - \sqrt{(\kappa_a x_i + \sin\theta_a)^2 + (\kappa_a y_i - \cos\theta_a)^2}\right\}^2 &= 0 \\ \frac{1}{\kappa_a^2}\left\{1 - \sqrt{(\kappa_a x_j + \sin\theta_a)^2 + (\kappa_a y_j - \cos\theta_a)^2}\right\}^2 &= 0\end{aligned} \tag{B1}$$

Next we change to polar coordinates $(r_i\cos\psi_i, r_i\sin\psi_i) = (x_i, y_i) \rightarrow (r_i, \psi_i)$, which gives us

$$\begin{aligned}\kappa_a &= \frac{2}{r_i}\sin(\psi_i - \theta_a) \\ \kappa_a &= \frac{2}{r_j}\sin(\psi_j - \theta_a)\end{aligned} \tag{B2}$$

Eliminating κ_a from eq. (B2) yields

$$\tan\theta_a = \frac{r_j \sin\psi_i - r_i \sin\psi_j}{r_j \cos\psi_i - r_i \cos\psi_j} \tag{B3}$$

References

[1] C. Peterson, "Parallel Distributed Approaches to Combinatorial Optimization Problems - Benchmark Studies on TSP", *Neural Computation* **2**, 261 (1990).

[2] C. Peterson and B. Söderberg, "A New Method for Mapping Optimization Problems onto Neural Networks", *International Journal of Neural Systems* **1**, 3 (1989).

[3] R. Durbin, and D. Willshaw, "An Analog Approach to the Traveling Salesman Problem Using an Elastic Net Method", *Nature.* **326**, 689 (1987).

[4] B. Denby, "Neural Networks and Cellular Automata in Experimental High Energy Physics", *Computer Physics Communications* **49**, 429 (1988).

[5] C. Peterson, "Track Finding with Neural Networks", *Nuclear Instruments and Methods* **A279**, 537 (1989).

[6] G. Stimpfl-Abele and L. Garrido, "Fast Track Finding with Neural Nets", UAB-LFAE 90-06 (1990) (submitted to Computer Physics Communications).

[7] L. Gislén, C. Peterson and B. Söderberg, "Rotor Neurons - Basic Formalism and Dynamics", LU TP 91-21 (submitted to Neural Computation).

[8] C. Peterson, "Neural Networks and High Energy Physics", *Proc. of International Workshop on Software Engineering, Artificial Intelligence and Expert Systems for High Energy and Nuclear Physics, Lyon Villeurbanne, France, March , 1990*, eds. D. Perret-Gallix and W. Woijcik, Editions du CRNS (Paris 1990).

[9] M. Gyulassy and H. Harlander, "Elastic Tracking and Neural Network Algorithms for Complex Pattern Recognition", *Computer Physics Communications* **66**, 31 (1991).

[10] A. Yuille, K. Honda and C. Peterson, "Particle Tracking by Deformable Templates", *Proceedings of 1991 IEEE INNS International Joint Conference on Neural Networks*, Vol. 1, pp 7-12, Seattle, WA (July 1991).

[11] A.L. Yuille, "Generalized Deformable Models, Statistical Physics, and Matching Problems", *Neural Computation.* **2**, 1 (1990).

[12] K. Koffka, *Principles of Gestalt Psychology*, Harcourt, Brace and World, New York (1935).

[13] M. Wertheimer, *Laws of Organization in Perceptual Forms*, Harcourt, Brace and Co., London (1938).

[14] M. Regler and R. Frühwirth, "Reconstruction of Charged Tracks", *Proc. Advanced Study Institute on Techniques and Concepts in High Energy Physics, St. Croix*, Plenum, Rochester (1989).

[15] P.J. Huber, *Robust Statistics*, John Wiley and Sons, New York (1981).

[16] A.L. Yuille, Yang, T. and D. Geiger, "Robust Statistics, Transparency and Correspondence'", Harvard Robotics Laboratory Technical Report 90-7. 1990.

[17] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, John Wiley and Sons, New York (1973).

[18] H. Grote, "Pattern Recognition in High-Energy Physics". *Rep. Prog. Phys.* **50**, 473 (1987).

[19] C.Brand et al.,"The DELPHI Time Projection Chamber", *IEEE Trans. Nucl. Sci.* **36** 122 (1989); "DELPHI Event Generator and Detector Simulation User's Guide", DELPHI 89-67 PROG 142, CERN, July 1989;
"DELSIM Reference Manual", DELPHI 89-68 PROG 143,CERN, September 1989.

[20] R.D. De Veaux, *Parameter Estimation for a Mixture of Linear Regression.* PhD. Thesis. Dept. of Statistics. Stanford University. 1986.

[21] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, "Optimization by Simulated Annealing", *Science* **220**, 671 (1983).

[22] D. Geiger and A.L. Yuille, "A Common Framework for Image Segmentation", *International Journal of Computer Vision* **6**, 227 (1991).

[23] R. Durbin, R. Szeliski and A.L. Yuille, "An analysis of the Elastic Net Approach to the Traveling Salesman Problem", *Neural Computation* **1**, 382 (1989).

[24] Dempster, A.P., Laird, N.M. and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm". *Journal of the Royal Statistical Society* **B 39**, 1 (1977).

[25] See e.g. W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, *Numerical Recipes*, Cambridge University Press, Cambridge (1986).